Modelling Competitive Co-operation of Agents in a Compositional Multi-Agent Framework

Brazier, F.M.; van Eck, P.A.T.; Treur, J.

published in
Knowledge Acquisition, Modelling and Management, Proc. of the 10th European Knowledge Acquisition Workshop, EKAW'97,
1997

Modelling Competitive Co-operation of Agents in a Compositional Multi-Agent Framework

Frances Brazier, Pascal van Eck and Jan Treur

Department of Mathematics and Computer Science, Vrije Universiteit Amsterdam
De Boelelaan 1081a, NL-1081 HV Amsterdam, The Netherlands
URL: `http://www.cs.vu.nl/`, Email: `{frances,patveck,treur}@cs.vu.nl`

1 Introduction

In many multi-agent domains competitive agents need to co-operate. In this paper a generic model for competitive agents is introduced: a model which can be used to support the design of agents in diverse knowledge-intensive domains. An example of the type of multi-agent situation in which this model can be applied is one in which a number of agents wish to access a given (information) resource and explicit knowledge is available (to either the accessing agents, or the resource to be accessed, or all agents involved) on appropriate orderings or priorities between the transactions from the different agents. This model includes explicit knowledge of possible communication and co-operation strategies for individual agents, but also identifies the type of communication needed between agents and the interaction needed between agents and the material world, specified in the compositional multi-agent system modelling framework DESIRE [1,2].

DESIRE is briefly introduced in Section 2. In Section 3, the role of competitive co-operation in a competitive situation is discussed. A generic model of an arbitrary competitive agent is presented in Section 4. Section 5 describes a generic model for limited resource acquisition. In Section 6, a comparison is made with an algorithmic approach to limited resource access in the domain of operating systems (see for example [4]). Discussion and further research are presented in Section 7.

2 The Multi-Agent Modelling Framework DESIRE

The multi-agent compositional modelling framework DESIRE provides support for the design of a conceptual model of the behaviour of (interacting) agents. Compositional agent models define the structure of the architectures: components in a compositional model are directly related to agents and their tasks. Existing generic agent models can be used to design specific agent models. During analysis and design, relevant components in a generic model are refined by (1) more detailed analysis of the tasks of which such components are comprised and/or (2) inclusion of specific domain knowledge. The five types of knowledge represented in the DESIRE framework at a conceptual level, detailed level and at an operational level are:

- The compositional structure of agents and their tasks. Tasks can be composed or primitive and are characterised by their input and output knowledge structures;
- Interaction within and between agents and tasks;
- Temporal relations between tasks, represented by rules in a temporal logic;
- Delegation of tasks to agents;
- Knowledge structures.

The representation at the operational level is automatically generated from the representation at the detailed level.

Published in: E. Plaza and R. Benjamins (Eds.) *Knowledge Acquisition, Modeling and Management.* Lecture Notes in Artificial Intelligence, Vol. 1319, pp. 317-322, Springer-Verlag, 1997.

3 A Generic Agent Model

To model an agent capable of competitive co-operation, a generic model of an agent, developed in other multi-agent domains, is employed [6]. This model fulfils the four characteristics required for the weak notion of agency described in [7]: agents must be capable of (1) maintaining interaction with their environment by observing and performing actions in the world: *reactivity*; (2) taking the initiative: *pro-activeness*; (3) performing social actions like communication and co-operation: *social ability*; and (4) operating without the direct intervention of other (possibly human) agents: *autonomy*.

This generic agent model has six top-level components as shown in Figure 2. Interaction with the environment is performed by the components Maintain World Information and World Interaction Management. Social actions are managed by the components Agent Interaction Management and Cooperation Management. The agent's processes are co-ordinated by the component Own Process Control, enabling the agent to act autonomously and take the initiative if this is required. Tasks specific to the agent itself are included in the component Agent Specific Tasks. In most domains of application, these six components are further refined, as illustrated for competitive co-operation in Section 5 below. In general, refinement of a generic model involves specialisation (i.e., components within components are distinguished) and instantiation (i.e., (domain) specific instances of signatures and knowledge are defined). Agent models differ in the level of refinement required depending on the relative importance of the characteristics involved. Knowledge required to refine agent models includes (1) knowledge of an agent's priorities with respect to its processes, (2) knowledge of which and how information is exchanged with other agents and the external world, (3) knowledge of how information received from the external world and other agents is to be analysed and (4) knowledge of how co-operative an agent is in given situations in relation to other agents. (In [8], other approaches to agent-based knowledge modelling can be found).

Within DESIRE, agents are modelled as components that run concurrently, equipped with information links for inter-agent information exchange. Task control at the top level of a multi-agent system is minimal: it is most often restricted to the initiation of agents, an external world and the links between agents. As components, agents have their own internal agent task control. This task control knowledge specifies (asynchronous) information exchange to other agents and the world, and (concurrent) activation or suspension of internal components.

4 Competitive Agents in Co-operative Information Systems

In many real life situations, co-operation is an effective approach for allocation of limited resources. In this paper, housing is the limited resource used to illustrate the use of a generic model for competitive co-operation. Allocation of apartments within the city to individuals (with a monthly rent between x and y) is regulated by governmental policy. Two key characteristics of the policy are the role assigned to the real estate agents, namely that of co-ordination agents, and the use of a static priority scheme (duration of an individual's subscription).

Practice shows that groups of subscribed individuals co-operate—together they determine their chances and determine individual and group strategies. In deadlock situations the real estate agent is responsible for conflict resolution. To design an agent system to support an individual in need of an apartment, a model of the interaction and knowledge required to effectuate strategies such as those discussed above, is needed. This example is used below to illustrate the generic aspects of competitive co-operation.

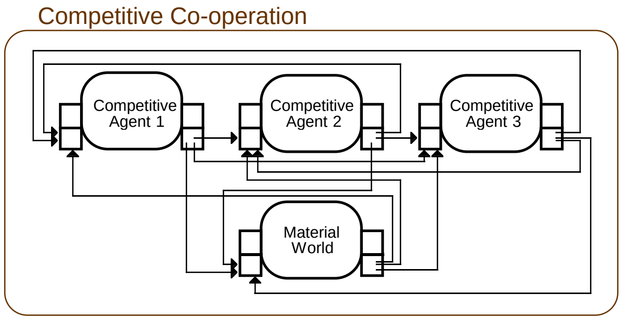

Fig. 1: Three agents and the material world.

5 A Generic Model of a Competitive Co-operative Agent

Competitive co-operation for resource allocation requires at least two agents and a material world. In Figure 1, three agents are depicted for the purpose of illustration. In this figure, the rounded boxes represent (composed or primitive) components, in this case agents. The small boxes attached to the components' sides, represent the agents' input and output interfaces. The links in Figure 1 between agents are defined to allow communication between agents; the links between the agents and the material world are defined to allow observations and actions to be performed. Task control is not depicted.

Figure 2 shows the composition of an individual agent and the information exchange between its components. In the following paragraphs, each of these components will be described. Due to space restrictions, it is not possible to provide more detail; instead the interested reader is referred to [5].

Maintain World Information The component Maintain World Information stores the information an agent has about the world state, namely presence of agents and resources. This information can be acquired by observation of the world (via World Interaction Management), communication with other agents, or (defeasible) reasoning.

Agent Specific Tasks In this generic agent model, only one agent specific task is modelled, namely Obtain Resource. Other agent specific tasks are not specified in the generic model (e.g. tasks in which the need to access a resource is determined).

Cooperation Management The component Cooperation Management consists of the following four components (see Figure 3):

- **Update Current Agent Information** All relevant information on other agents, often obtained by communication and/or (defeasible) reasoning, is maintained here.
- **Determine Access** Input facts about (1) the world; e.g., obtained by observation, communication, or (default or closed world) assumptions, (2) priorities between agents (received from the component Determine Priority) and (3) co-operativeness of other agents (received from the component Determine Cooperation), are used to analyse a world state (no matter how it was reached) and to draw conclusions about access to a resource.

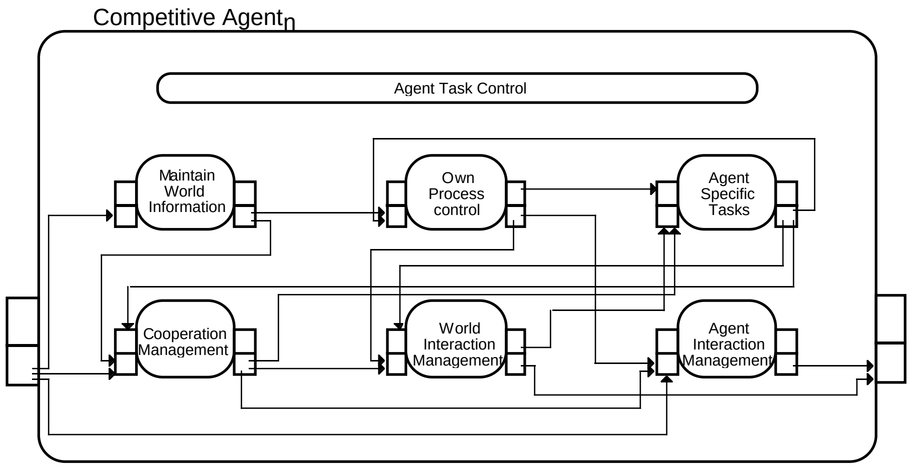

Fig. 2: Agent task composition and information exchange.

Most of the knowledge used in Determine Access specifies the conditions under which access to the resource will not be granted. The two rules shown below determine whether there is a conflict in the sense that another agent and the agent itself are both interested in accessing the resource. Note that the knowledge specified in this knowledge base does not refer directly to the application domain described in Section 4. It is generic in the sense that it can in principle be used for all domains in which limited access to resources plays a role.

if wants_resource(A) **and** wants_resource(self) **then** conflicting_needs(A,self)
if conflicting_needs(A,self) **and** has_priority_over(A,self) **then** access_blocked_by(A)

- **Determine Priority** The task of the component Determine Priority is to determine which of two agents may access the resource first, if a conflict exists. The instantiation of Determine Priority contains domain specific knowledge, such as knowledge required for the domain outlined in Section 4. It is important to note that this knowledge may not always be sufficient to derive a unique conclusion: there may be circumstances in which no priority can be assigned.
- **Determine Cooperation** Knowledge of which agents are willing to co-operate with which other agents acquired through observation and reasoning, is used by this component to determine the level of co-operation. In the generic agent model, three options for static facts about the agent self are specified: one for a shy agent, one for a bold agent and one for a moderate agent:

if concluded(no_access_decision) **and** modest(self) **then** to_communicate_to(A,ok)

World Interaction Management, **Agent Interaction Management** and **Own Process Control** The tasks of the component World Interaction Management are (1) to perform observations (including observation of the presence and relevance of other agents) and (2) to perform the action proposed by Obtain Resource in the component Agent Specific Tasks. The component Agent Interaction Management manages communication between an agent and other agents. The role of the component Own Process Control is to determine which information is needed to decide whether access to a resource is allowed, and where this information is to be found.

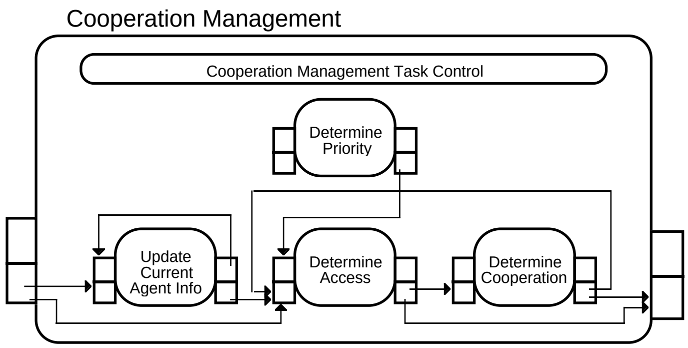

Fig. 3: Composition of Cooperation Management.

Agent task control The component Own Process Control determines which information is needed to decide whether access to a resource is allowed and where this information is to be found. Task control knowledge specifies activation of the component Own Process Control in one of the following ways: (1) The component Agent Interaction Management notices that one of the other agents requests attention, for example to access a resource, and notifies Own Process Control of the fact that new information has been received. On the basis of this new information, Own Process Control may decide that Cooperation Management is to be activated to determine whether access is to be granted. (2) The component Agent Specific Tasks expresses a need for information to the component Own Process Control, which decides which specific information is needed and where it is to be found. If Own Process Control recognises the need to access a limited resource, the component Cooperation Management is activated first to determine whether access is allowed. If access is allowed, Obtain Resource is activated.

6 Comparison with an Algorithmic Approach

The specification of the task model presented in this paper can be compared to approaches to mutual exclusion problems in more conventional environments, such as the algorithm described by Ricart and Agrawala [4]. Ricart and Agrawala's algorithm assumes the following conditions hold: (1) each agent notices each other agent's presence; (2) communication never fails; (3) each agent has the same *complete* knowledge of priorities between agents and (4) if agent A has higher priority than agent B, agent B is assumed to communicate that it grants access to the resource to agent A.

If these conditions hold, the task model described in this paper specifies the same process as the algorithm. The conditions describe a rather strictly defined domain of application, as found in, for example, the domain of operating systems. For less strictly defined real world domains, however, incompleteness of observations, defeasible communications, incomplete knowledge of priorities, inconsistencies between conclusions drawn by different agents, uncooperative agents, etc. are most common. The compositional model introduced here is particularly suitable in these domains for the following reasons: (1) by virtue of the reflective structure of the model, different strategies can be modelled with minimal effort, (2) assumptions with respect to for instance communication, priorities and co-operation appear explicitly in the model, and (3) the different types of knowledge are explicitly distinguished. The distinction between different types of knowledge and different types of behaviour results in flexibility, adaptability and transparency: essential characteristics of a knowledge-based approach.

7 Discussion and Future Research

In this paper, a generic model is presented for competitive agents in a domain that requires limited access to a given resource. Modelling the specific types of knowledge involved and the behaviour of agents in relation to each other results in a transparent compositional model. One specific domain of co-operation has been used, in which resource access is granted first on the basis of (given) priorities, and (if no decision can be taken) on the basis of dynamically observed co-operativeness of other agents. However, the modularity of the architecture, in which both static and dynamic behaviour are explicitly specified, allows for flexible adaptability to other strategies. Current research focuses among others on the further development of the semantics of the framework ([2]), on verification and validation ([3]) and on modelling beliefs, desires, intentions and commitments in multi-agent systems.

Acknowledgements This research was partly supported by ESPRIT III BRA project 6156 DRUMS II on Defeasible Reasoning and Uncertainty Management Systems.

References

1. Brazier, F.M.T., Treur, J., Wijngaards, N.J.E., and Willems, M. Formal specification of hierarchically (de)composed tasks. In: Gaines, B.R. and Musen, M. (eds.), *Proc. of the 9th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*, Vol. 2, pp. 25/1-25/20, SRDG Publications, Deptartment of Computer Science, University of Calgary, 1995.
2. Brazier, F.M.T., Treur, J., Wijngaards, N.J.E., and Willems, M. Temporal semantics of complex reasoning tasks. In: Gaines, B.R. and Musen, M. (eds.), *Proc. of the 10th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*, Vol. 1, pp. 15/1-15/17, SRDG Publications, Deptartment of Computer Science, University of Calgary, 1996. Extended version to appear in *Data and Knowledge Engineering*.
3. Cornelissen, F., Jonker, C. and Treur, J. Compositional verification of knowledge-based systems: A case study for diagnostic reasoning. In: *Proc. European Knowledge Acquisition Workshop EKAW'97*, Springer-Verlag, 1997 (this volume).
4. Ricart, G. and Agrawala, A.K. An optimal algorithm for mutual exclusion in computer networks. *Comm. ACM*, Vol. 24(1), pp. 9-17, Jan. 1981.
5. Brazier, F.M.T., Van Eck, P.A.T. and Treur, J. Modelling cooperative behaviour for resource access in a compositional multi-agent framework. In: Fiadeiro, J.L. and Schobbens, P.-Y. (eds.) Proc. 2nd Workshop of the ModelAge Project, ModelAge'96, Universidade de Lisboa, 1996.
6. Brazier, F.M.T., Dunin Keplicz, B., Jennings, N. and Treur, J. DESIRE: Modelling Multi-Agent Systems in a Compositional Formal Framework, *International Journal of Cooperative Information Systems*, Vol. 6, Huhns, M. and Singh, M. (eds.), Special Issue on Formal Methods in Cooperative Information Systems: Multi-Agent Systems, 1997.
7. Wooldridge, M.J. and Jennings, N.R. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115-152, 1995.
8. Dieng, R., Corby, O. and Labidi, S. Agent-based knowledge acquisition. In: Steels, L., Schreiber, G. and Van de Velde, W. (eds.) *A Future for Knowledge Acquisition. Proc. EKAW'94.* LNAI, Vol. 867, pp. 63-82, Springer-Verlag, 1994.